REMARKS/ARGUMENTS

This Amendment and the following remarks are intended to fully respond to the Office Action mailed September 13, 2005. In that Office Action, claims 1-13 were examined, and all claims were rejected. More specifically, claims 1-13 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Lee (USPN 6,732,362), hereinafter "Lee," in view of Petty et al. (USPN 6,342,907), hereinafter "Petty." Reconsideration of these rejections, as they might apply to the original and amended claims in view of these remarks, is respectfully requested.

In this Response, claim 11 has been amended; no claims have been canceled or added.

Claim Rejections - 35 U.S.C. § 103(a)

Claims 1-13 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Lee in view of Petty. Applicants respectfully traverse the rejection.

To establish *prima facie* obviousness under 35 U.S.C. 103(a), three basic criteria must be met, namely: (1) there must be some suggestion or motivation to combine the references or modify the reference teaching; (2) there must be a reasonable expectation of success; and (3) the reference or references when combined must teach or suggest each claim limitation. MPEP § 2142. Applicants respectfully assert that the Examiner has failed to establish or the amended claims preclude a finding of a *prima facie* case of obviousness because the reference fails to disclose or suggest all of the limitations of the pending claims.

Lee provides "an object oriented exchange managing system and a method for installing an exchange resource." Col. 1, lines 51-53. More particularly, Lee provides a system and method for generating an object instance having certain, set attributes from known classes that provide a resource from an exchange service. *See generally*, col. 3, lines 5-60. Lee differs from the present invention in many important ways.

First, Lee does not include the same components as the present invention, i.e., a schema document that defines a property sheet (hereinafter referred to as the property sheet), a property sheet definition, a schema document that defines a property page (hereinafter referred to as the property page), and a property page definition. Lee simply discloses an object-oriented ODBMS. *See* col. 2, lines 35-42.

Applicants would first like to note that the Examiner has not identified any particular item in Lee that supposedly discloses either the property sheet definition or the property page definition. In fact, Applicants cannot locate any material in Lee that remotely applies to the property sheet definition or the property page definition. In the present invention as defined by the claims, the property sheet conforms to a property sheet definition. The application states, "[t]he method initially involves receiving a first schema document that conforms to a property sheet definition, such as an <u>XML document type definition</u>, such that the first schema document defines a property sheet." Page 5, lines 16-19; *see also* page 16 lines 21-22. The property page conforms to a property page definition. The application states:

[T]he property pages may be based on an XML schema, which defines both the types of controls that a property page will contain and the layout of those controls on a page. Table 2 illustrates an example of a Document Type Definition or DTD that defines an XML schema for creating a property page.

Table 2: Example DTD for an XML Schema for a Property Page

<!ELEMENT propertySheets	(propertySheet+)>
<!ELEMENT propertySheet	propertySheetID, name, description, getDataHandler, (setDataHandler,propertyPage+)>
<!ELEMENT propertyPage	(propertyPageID, name, description, attributes,layout)>
<!ELEMENT attributes	(attribute+)>
<!ELEMENT attribute	(attributeID, name, defaultValue?, displayHints?)>
<!ELEMENT layout	(row+)>
<!ELEMENT row	(item+)>
<!ELEMENT item	(attributeID?, displayHints)>

<!ELEMENT displayHints	(Type, (Text | Label)?, rows?, cols?, Size?, Show)>
<!ELEMENT Label	(#PCDATA)>
<!ATTLIST Label	width CDATA "20%" style (left|right|top) "left">
<!ELEMENT Text	(#PCDATA)>
<!ELEMENT rows	(#PCDATA)>
<!ELEMENT cols	(#PCDATA)>
<!ELEMENT propertySheetID	(#PCDATA)>
<!ELEMENT propertyPageID	(#PCDATA)>
<!ELEMENT attributeID	(#PCDATA)>
<!ELEMENT name	(#PCDATA)>
<!ELEMENT description	(#PCDATA)>
<!ELEMENT getDataHandler	(#PCDATA)>
<!ELEMENT setDataHandler	(#PCDATA)>
<!ELEMENT defaultValue	(#PCDATA)>
<!ELEMENT Type	(#PCDATA)>
<!ELEMENT Size	(#PCDATA)>
<!ELEMENT Show	(#PCDATA)>

Page 29, line 23 – page 24, line 4.

Lee does not contain any mention of any Document Type Definition (DTD). The Microsoft Computer Dictionary, Fifth Edition, Copyright 2002, defines a DTD as "[a] separate document that contains formal definition of all of the data elements in a particular type of HTML, SGML, or XML document, such as a report or a book. By consulting the DTD for a document, a program called a parser can work with the markup codes that the document contains." The DTD, and in kind, the property sheet definition, and the property page definition are simply different than a class, as understood in the art. By understanding the differences between the property sheet definition and the property page definition, claimed in the present invention, and the classes described in Lee, other problems with Lee and Lee's inapplicability to the present invention become clearer.

Second, the first schema document and the second schema document(s) cannot be understood as objects. Examiner has read the "base class" on the property sheet and an object instance of the base class on the property page. *See* Page 3, line 6-11. Thus, Examiner purports

that the property sheet is a base class and each property page is an instance object of the property sheet class. This interpretation is improper and thus unfortunate as it leads to many of problems when using Lee as part of an obviousness rejection.

The property page is not an instance of the property sheet, but an attachment or modification of the property sheet. The application states:

> Upon receiving the property page, a determine operation 604 determines whether the parent object, i.e., property sheet for that property page has already been defined by another resource. If so, the flow branches YES to append [*sic*] operation 606.
>
> The append operation 606 <u>appends the received property page to the property sheet</u>. Essentially, the property sheet definition is modified, such as by the property page manager to include a pointer to the new property page. Thus, the next time the property sheet is called, the new property page information is displayed along with other property pages for the supported object. Page 33, line 19 – page 34, line 8 (emphasis added).

In addition, the interpretation that the property sheet is a class and the property page is an instance object of the class reads out parts of the claims. The property sheet conforms to a property sheet definition. Therefore, the property sheet does not necessarily define itself, as with a "base class." Also, the property page conforms to a property page definition, and, thus, the property page does not conform to the property sheet or "inherit" from the property sheet.

Further, the property sheet and property pages are not objects, but represent objects. As the applications states with reference to Fig. 5:

> With respect to certain aspects of the present invention, the property sheets that are exposed to the system 304 by one resource are extendible by other resources. Fig. 5 illustrates the concept of having a separate, independent application or resource extend an existing property sheet. In Fig. 5, <u>a property sheet representing a particular user object is illustrated in display 500</u>. Consequently, the display 500 represents the object itself. The object provides a title bar 502, an active region 504, and a scope list 506. The title displays the title of an object as defined by the property sheet. The active region 504 displays controls and data fields for one property page, as selected from the list 506. The list 506 lists the various property pages that may be displayed in the active region 504 that relate to the user object 500 (emphasis added).

As shown in Fig. 5, the personal information property page control 508 has been selected and therefore the active region 504 displays the property page relating to the personal information for the particular user. Other selection controls for additional property pages are displayed in the scope list 506, such as a job information control 510 and an email information control 512. Additional information about the visual representation of the schema at the client computer system is displayed below with respect to the management console.

Independent resources or applications may define one or more property pages associated with a particular object, such as user object 500 shown in Fig. 5. Page 28, line 12 to page 29, line 7 (emphasis added).

Lee creates objects not documents representing objects. Lee states:

A Scoping Function Processor (SFP) **108** analyzes an EXCHANGE RESOURCE REQUEST message received from an operator via the UIP **106** to determine the scope of a class to be processed from a base class. A Filtering Function Processor (FFP) **110** processes an attribute value within the class that satisfies the operator-requested condition. A Resource Installation Processor (RIP) **112** generates an object instance representative of information about the requested exchange resource and stores the object instance in an Object Data Base Management System (ODBMS) **116**. Col. 2, lines 52-62 (emphasis added).

Third, the present invention as defined in the claims and Lee simply function differently.

The present invention as defined in the claims creates a property sheet if necessary. *See* page 33, line 19 – page 34, line 2 ("Upon receiving the property page, a determine operation 604 determines whether the parent object, i.e., property sheet for that property page has already been defined by another resource."). Then a property page, related to a resource and conforming to property page definition, is received. *See* page 33, lines 17-18 ("... receive operation relates to the property sheet manager 332 receiving a property page..."). The property page is appended to the created property sheet. *See* page 34, lines 4-6 ("The append operation 606 appends the received property page to the property sheet. Essentially, the property sheet definition is modified, such as by the property page manager to include a pointer to the new property page.") Each property sheet relates to an object, such as a user object, i.e., there can be several property sheets because there will be a property sheet for every object that uses the resource and a corresponding property page relating to the resource. *See* page 28, lines 15-19 ("With respect to

certain aspects of the present invention, the property sheets that are exposed to the system 304 by one resource are extendible by other resources. Fig. 5 illustrates the concept of having a separate, independent application or resource extend an existing property sheet. In Fig. 5, a property sheet representing a particular user object is illustrated in display 500.").

Lee is different because Lee discloses a set of classes for the resources. *See* col. 3, lines 14-17 ("... the managing system **100** analyzes the received message using the SFP **108** to determine whether there is a base class corresponding the requested resources model in step **202.**") (emphasis added). Lee then checks if the available classes support the new resource. *See* col. 3, lines 17-21 ("To this end, the SFP **108** determines whether the requested resource is available or whether the exchange **(104/105)** is capable of providing the service corresponding to the requested resource (i.e., whether the exchange is equipped to provide the smart card system."). Lee does not create new classes if the current classes do not support the new resource. *See* col. 3, lines 24-27 ("In the absence of such base class, the managing system **100** transmits an error message indicating a wrong resource installation request back to the operator through the GUI **102** in step **204.**"). If the class does support the new resource, an attribute is determined and an object instance, with the changed attribute, is created for the new resource. The new resource is added to the ODBMS. Lee states:

> On the other hand, in the presence of the base class in step **202**, the managing system **100** determines the scope of a class to be processed by the SFP **108** in step **206**, and determines whether the attribute values of the class satisfy the operator-requested condition through the FFP **110** in step **208**. That is, the FFP **110**, in step **208**, determines one of the attribute[s] corresponding to the determined class, as each class has a set of attributes which defines different properties of the class. In step **210**, REP **112** generates an object instance to access and update the ODBMS and also sends the object instance to the exchange. Col. 3, lines 28-38 (emphasis added).

Lee is different because Lee does not receive new property pages for new resources and attach those new property pages to existing property sheets. Rather, Lee generates new objects

10

from existing classes and adds the new objects to the ODBMS. Lee also does not generate new property sheets if no property sheet exists. Lee only uses existing classes to generate objects. Also, in Lee, there is one object for each resource regardless of the number of users for the resource. For at least the reasons explained above, Lee does not disclose elements of the claimed invention as argued by the Examiner.

Petty does not overcome the inadequacies of Lee. Petty discloses a specification language to define user interface panels. *See* Abstract and col. 3, lines 22-25. The specification language is a Panel Definition Markup Language (PDML) that allows a user to specify the exact location of components displayed in the panel. *See* Abstract and col. 3, lines 25-30. In addition, Petty discloses a graphical editor that allows the creation and modification of platform-independent user interface panels; a conversion tool that can convert platform-specific user interface panels to corresponding platform-independent user interface panels; and a help generator tool that facilitates the generation of context-sensitive help for a user interface panel. *See* Abstract and col. 3, lines 30-37.

Petty does not disclose, anywhere in the patent, a property sheet, property page, property sheet definition, or a property page definition. Like Lee, Petty describes object-oriented systems but does not describe documents or DTDs. *See* col. 4, line 31—col. 5, line 15. Finally, Petty also does not disclose appending property pages to a property sheet when adding new resources to a system. Thus, neither Lee nor Petty, either alone or in combination, disclose the present invention as defined in the claims.

For at least the reasons given above, claim 1 and amended claim 11 are allowable over Lee and Petty either alone or in combination. All other claims, i.e., claims 2-10 and 12-13 depend from allowable claims 1 and 11 and are also allowable over Lee and Petty either alone or

in combination. As such, Applicants respectfully request that Examiner allow the claims and issue a Notice of Allowance at his earliest convenience.

Conclusion

This Amendment fully responds to the Office Action mailed on September 13, 2005. Still, that Office Action may contain arguments and rejections and that are not directly addressed by this Amendment due to the fact that they are rendered moot in light of the preceding arguments in favor of patentability. Hence, failure of this Amendment to directly address an argument raised in the Office Action should not be taken as an indication that the Applicants believe the argument has merit. Furthermore, the claims of the present application may include other elements, not discussed in this Amendment, which are not shown, taught, or otherwise suggested by the art of record. Accordingly, the preceding arguments in favor of patentability are advanced without prejudice to other bases of patentability.

It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

In light of the above remarks and amendments, it is believed that the application is now in condition for allowance and such action is respectfully requested.

Respectfully submitted,

Dated: 12/13/05
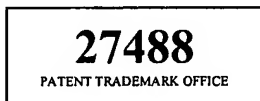
27488
PATENT TRADEMARK OFFICE

Tadd F. Wilson, Reg. # 54,544
MERCHANT & GOULD P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903
303.357.1648

12